X.509 Policy Processing Fix:

An Alternate Approach

May 13, 1999

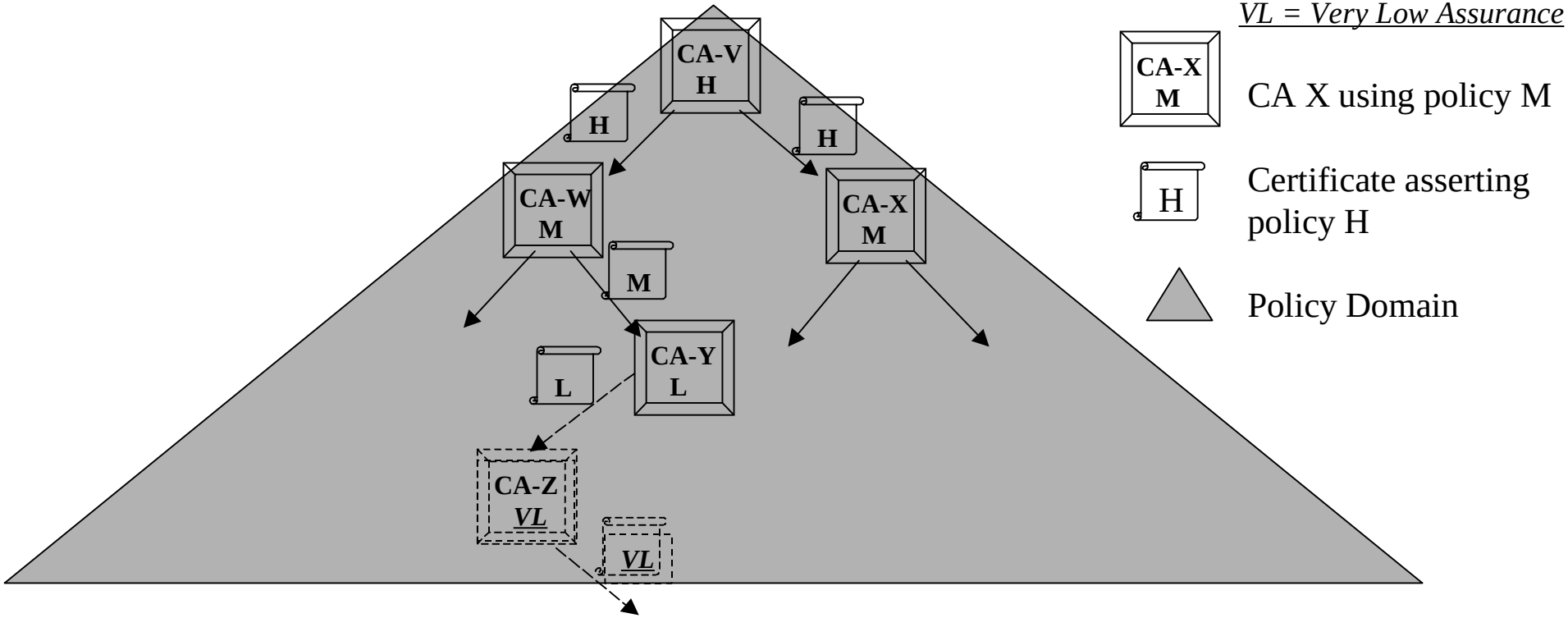
Sarbari Gupta (CygnaCom)

CygnaCom Solutions, Inc. ◆ Suite 100W, 7927 Jones Branch Drive, McLean, VA 22102 ◆ (703) 848-0883

Deployment Scenario 1

Business Goals: Establishment of policy domains where:

- superior CAs can apply higher assurance policies than subordinate CAs
- superior CAs can restrict the policies asserted by subordinate CAs
- path processing succeeds even when superior CAs do not assert a superset of all subordinate polices, but assert just the local policy
- subordinate policies (such as VL) can be added dynamically to a policy domain without affecting superior certificates

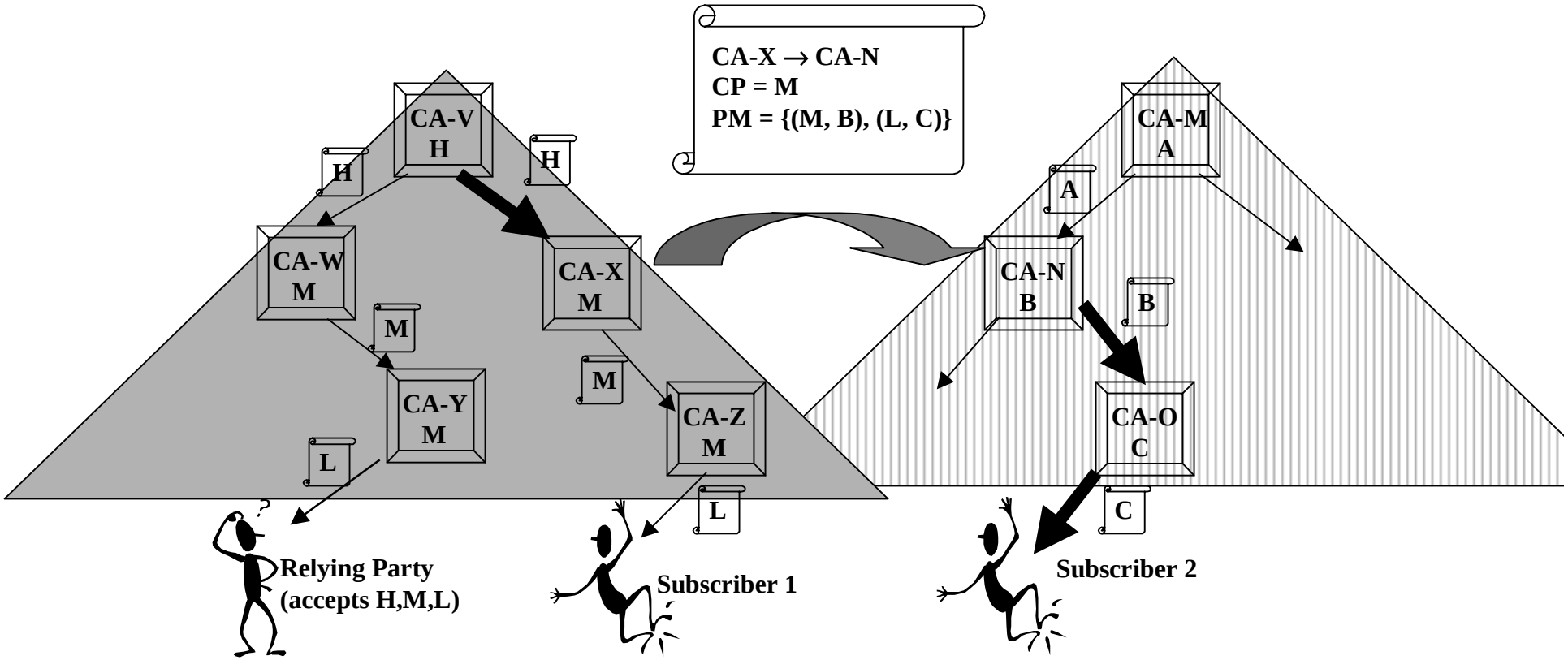


Deployment Scenario 2

Business Goals:

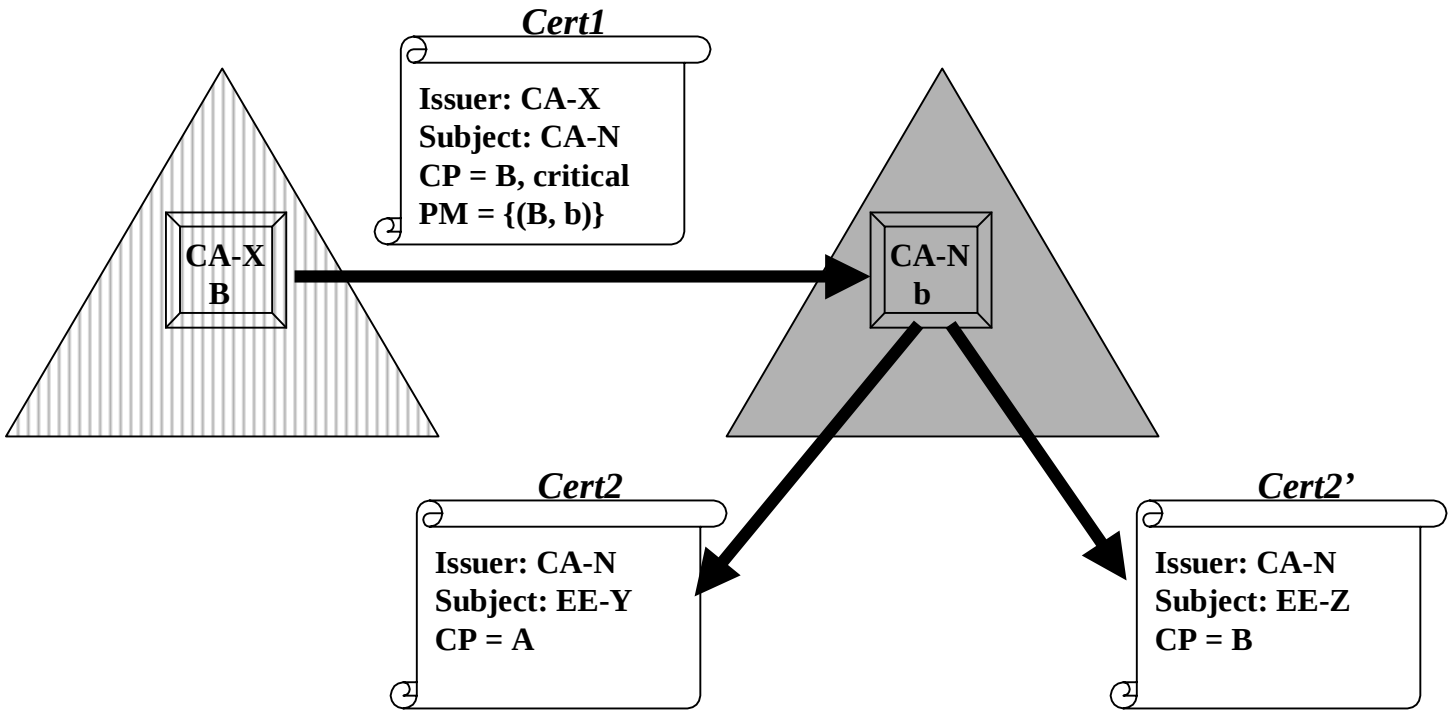
• Cross-certifying CAs can assert all possible policy equivalencies between their respective domains

• Relying party can authenticate Subscriber 2 as well as Subscriber 1 using certificate path via cross-certificate



Flaw Scenario 1 in X.509

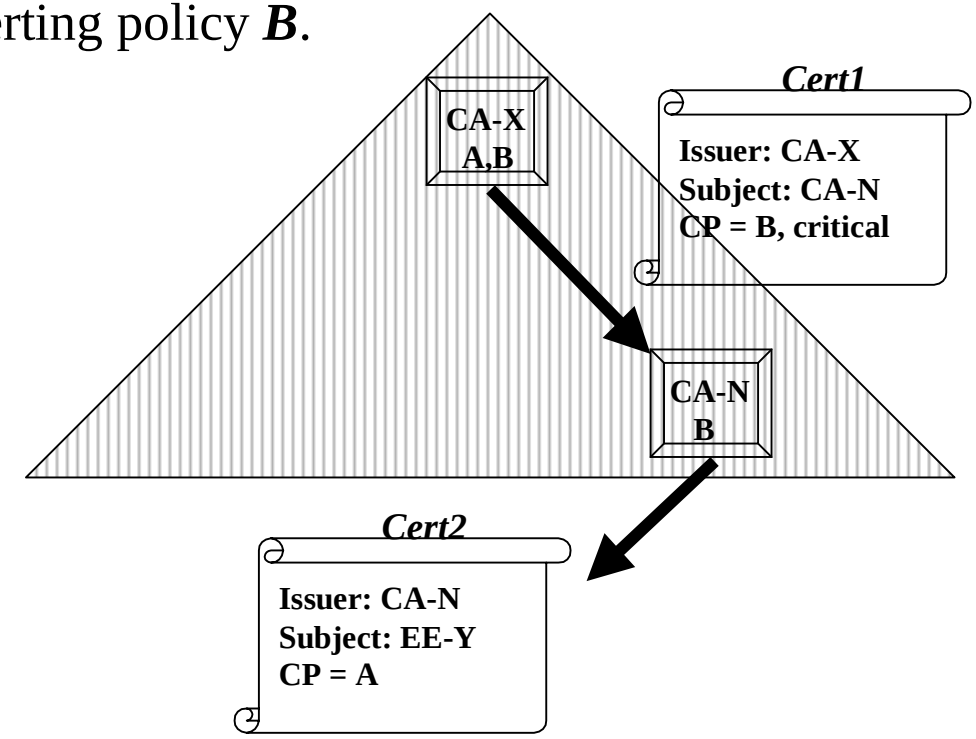
It appears to be difficult for a CA issuing a cross-certificate to a subject CA to restrict the policies that may be asserted by the subject CA.



If *initial-policy-set* is {A, B}, then the above chains will be valid. It is difficult for ***CA-X*** to restrict ***CA-N*** to only asserting policy ***b*** in ***Cert2*** and ***Cert2'***

Flaw Scenario 2 in X.509

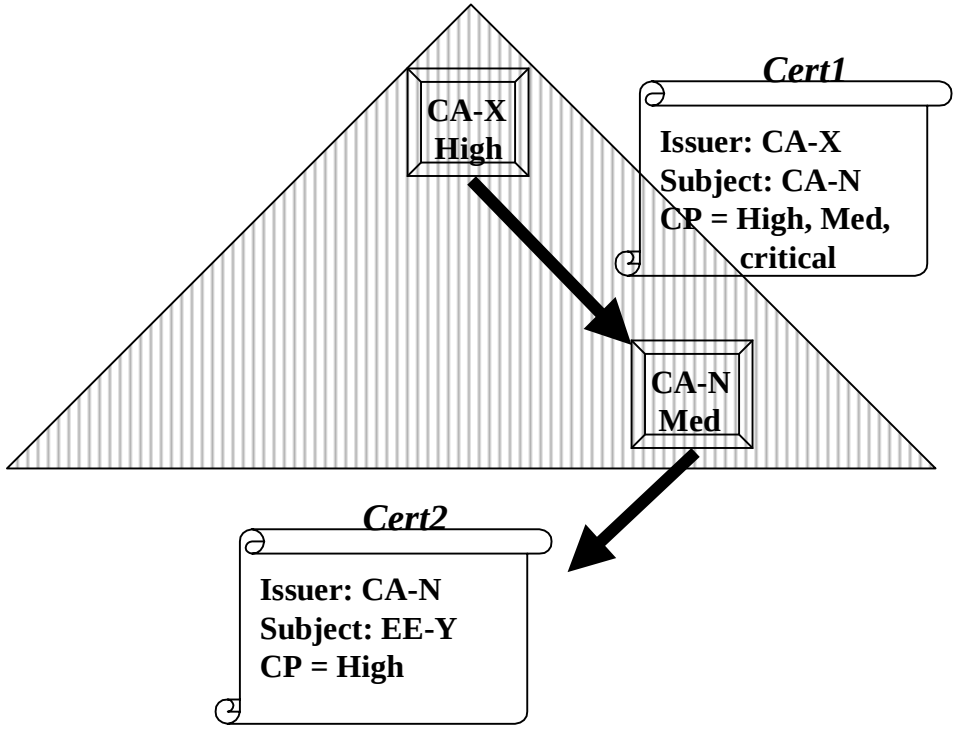
There appears to be no way for a superior CA to restrict the policies that may be asserted by a subordinate CA. Assume ***CA-X*** would like to restrict ***CA-N*** to only asserting policy ***B***.



If *initial-policy-set* is {A, B}, then the above chain will be valid. It is difficult for ***CA-X*** to restrict ***CA-N*** to only asserting policy ***B*** in ***Cert2***

Flaw Scenario 3 in X.509

It appears to be difficult for a superior CA (***CA-X)*** to restrict a subordinate CA (***CA-N)*** from asserting policy ***High***.



If *initial-policy-set* is {High, Med}, then the above chain will be valid.

Underlying Policy Flaws in X.509

All 3 flaw scenarios are symptomatic of two underlying flaws:

1) The *certificatePolicies* extension is overloaded to signify:
   – policies under which certificate was issued, indicating the purposes for which it may be used
   – polices that may be asserted by subordinate CAs, through use of the “critical” flag

2) The path processing logic does not check policies asserted in a cert against *authority-constrained-policy-set*

Fix to Policy Flaws in X.509

• Restrict the usage of the *certificatePolicies* extension to signify:
  – policies under which certificate was issued
  – criticality flag has no effect on path processing

•Add a new extension, *permittedPolicies* to signify
  – polices that may be asserted by subordinate CAs

• Update path processing logic such that asserted policies are always checked against *authority-constrained-policy-set*

Existing X.509v3 Extensions

certificatePolicies: assertion of the policies under which the certificate was issued, indicating the purposes for which it may be used.

policyMappings: asserted in CA certificates to express equivalency relations between policies in the issuer CA and subject CA policy domains

policyConstraints: Two separate fields -

- ***inhibitPolicyMapping*** field - disallows further policy mappings
- ***requireExplicitPolicy*** - not needed

New X.509v3 Extensions

• ***permittedPolicies***: populated in a CA certificate to restrict the set of policies that may be asserted by subordinate CAs

Updates to Path Processing (I)

- Inputs : No change
- Outputs : No change
- State variables : No change
- Initialization Step: No change

Updates to Path Processing (II)

- Processing of all certificates :

  c) If *explicit-policy-indicator* is set, …, check that at least one member of the *user-constrained-policy-set* appears in the certificate policies field. [NO CHANGE]

  d) If the *authority-constrained-policy-set* is not *any-policy*, check that the certificate policies extension is present, and that at least one member of the *authority-constrained-policy-set* appears in the certificate policies field. [UPDATE TO EXISTING STEP]

Updates to Path Processing (III)

- Processing of intermediate certificates :

  e) If policy-mapping-inhibit-indicator is not set: [NO CHANGE]

  – process any policy mapping extension with respect to … *user-constrained-policy-set* and add appropriate policy identifiers ...

  – process any policy mapping extension with respect to … *authority-constrained-policy-set* and add appropriate policy identifiers …

  f) If the ***permittedPolicies*** extension is present, compute the intersection of the policies in that extension and the *authority-constrained-policy-set* and put the result as the new value of the *authority-constrained-policy-set* [NEW STEP ADDED]

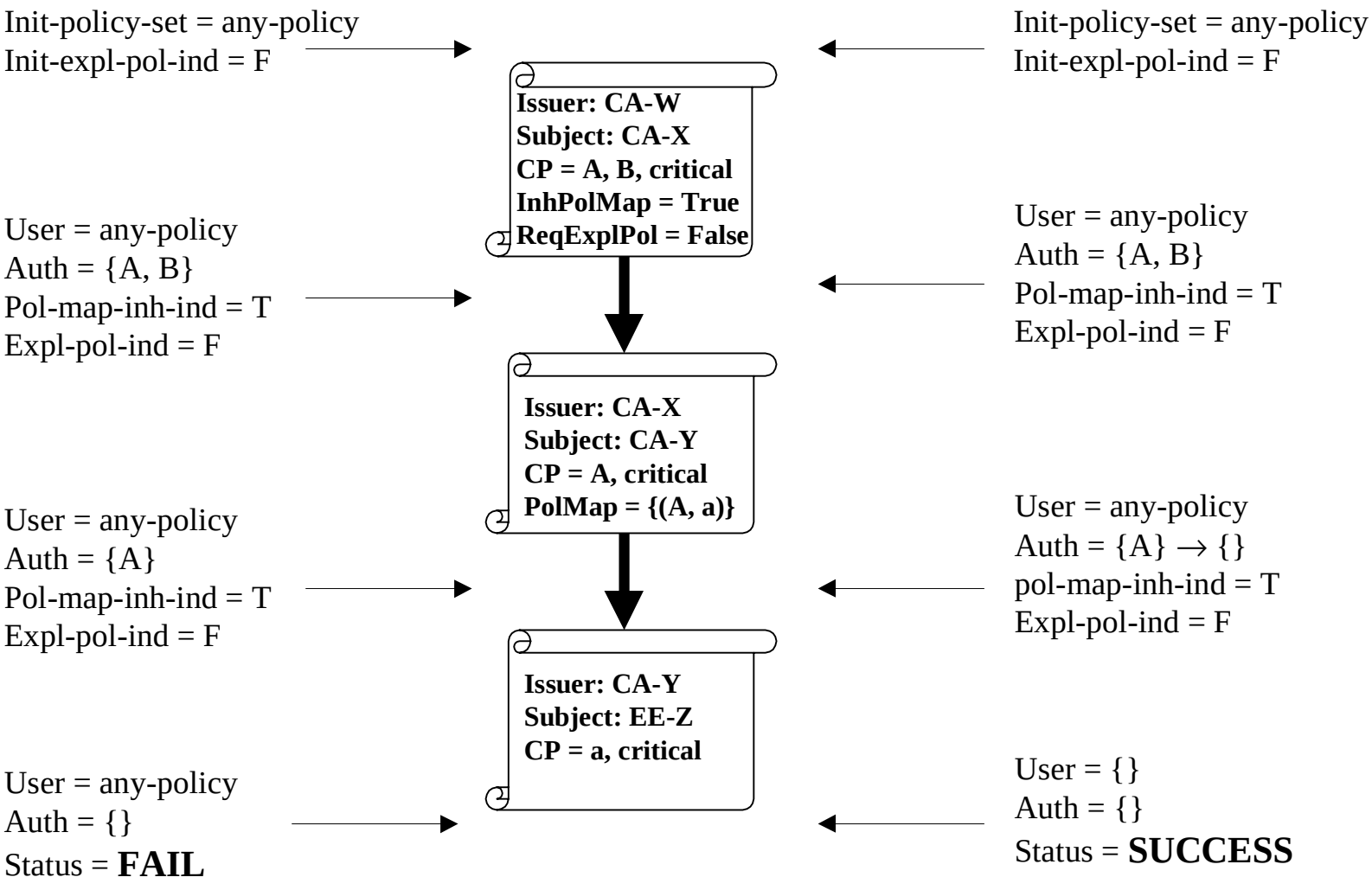
Advantages of New Approach

- Semantics of all existing extensions remain unchanged
- No restrictions on the way policies may be deployed within policy domains, when new extension is not used
- Minimal changes (through augmentation rather than replacement) in path processing algorithm
- Full backward compatibility with existing CAs and issued certificates
- Ability of superior CAs to restrict policies that may be asserted by subordinate CAs

Implications of Original Fix

- Semantics of certificate policies extension and policy mapping extension have changed
- A common policy OID MUST appear in every certificate in a valid chain => deployment restrictions
- CP extension must contain a superset of all policies used by subordinate CAs => deployment restrictions
- The certificate policies extension MUST be present, else cert chain will fail policy processing checks
- If policy mapping extension is non-critical, and relying party ignores it, certain unexpected chains pass validation
- Some certificate paths that would have failed the existing X.509 policy checks will succeed proposed policy checks (see next slide)

Implication of Original Fix



CURRENT STANDARD

ORIGINAL FIX TO STANDARD

Usage of certificatePolicies Extension

- This extension should include policy identifiers only for the policies that were used in issuing the subject certificate. For example, in Scenario 1, CA V should assert only {H}, instead of {H,M,L}
- The semantic description of this extension should be changed to "… policy information terms indicate the policy under which the certificate has been issued indicating the suitability of the certificate for specific purposes and applications"

Usage of permittedPolicies Extension

- When dynamic addition of lower assurance policies is desirable within a policy domain, the permittedPolicies extension should not be used within CA certificates within the domain.
- When the assertion of policies by subordinate CAs within a domain is to be restricted, the permittedPolicies extension should be populated with ALL subordinate policies that may be asserted be subordinate CAs.

Usage of PolicyMappings Extension

- This extension should be used to convey policy equivalence relations between two policy domains. Thus, when cross-certifying between policy domains, the policyMappings extension may include all possible equivalency statements between policies in the subject domain and issuer domain. This extension is not required to be limited to equivalency relations corresponding to the policies asserted within the cross-certificate.